

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-240493

(43)Date of publication of application : 11.09.1998

(51)Int.Cl.

G06F 3/16
G06F 13/00
G10L 3/00

(21)Application number : 10-001106

(71)Applicant : TEXAS INSTR INC <TI>

(22)Date of filing : 06.01.1998

(72)Inventor : HEMPHILL CHARLES T
BAKER SCOTT C

(30)Priority

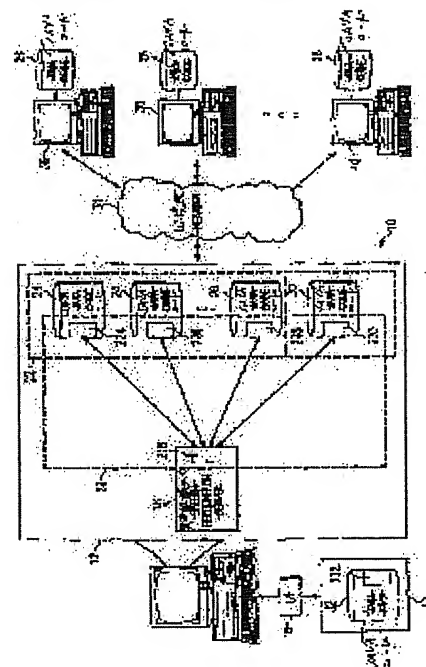
Priority number : 97 34642 Priority date : 06.01.1997 Priority country : US

(54) SYSTEM AND METHOD FOR ADDING VOICE RECOGNITION CAPABILITY TO JAVA

(57)Abstract:

PROBLEM TO BE SOLVED: To obtain a system that adds a voice recognition capability to Java by dynamically designating a grammar to an application program interface and making the application program interface communicate the grammar to a voice recognition server.

SOLUTION: A server side API 218 and an application side API 224 are collectively called an application program interface (API) 20. An application 24 designates one or more grammars to the API 20 during operation. A grammar is a set of rules that define a syntax and a vocabulary, and the application 24 responds to it. The API 20 communicates a grammar to a voice recognition server (SRS) 18, and the SRS 18 receives and stores the grammar. The SRS 18 receives a voice input after processing the grammar, also carries out voice recognition based on a grammar that is designated by a Java application 24 and generates a result.



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-240493

(43)公開日 平成10年(1998)9月11日

(51)Int.Cl.⁶
 G 0 6 F 3/16
 13/00
 G 1 0 L 3/00

識別記号
 3 2 0
 3 5 1
 5 5 1

F I
 G 0 6 F 3/16 3 2 0 H
 13/00 3 5 1 E
 G 1 0 L 3/00 5 5 1 Z

審査請求 未請求 請求項の数2 O L (全 12 頁)

(21)出願番号 特願平10-1106

(22)出願日 平成10年(1998)1月6日

(31)優先権主張番号 0 3 4 6 4 2

(32)優先日 1997年1月6日

(33)優先権主張国 米国 (U S)

(71)出願人 590000879

テキサス インスツルメンツ インコーポ
レイテッドアメリカ合衆国テキサス州ダラス, ノース
セントラルエクスプレスウェイ 13500

(72)発明者 チャールズ ティー. ヘムフィル

アメリカ合衆国テキサス州アレン, ホロー
リッジ ドライブ 735

(72)発明者 シー. スコット ベイカー

アメリカ合衆国時ケンタッキー州ベルサイ
ユ, ジョージア ストリート 105

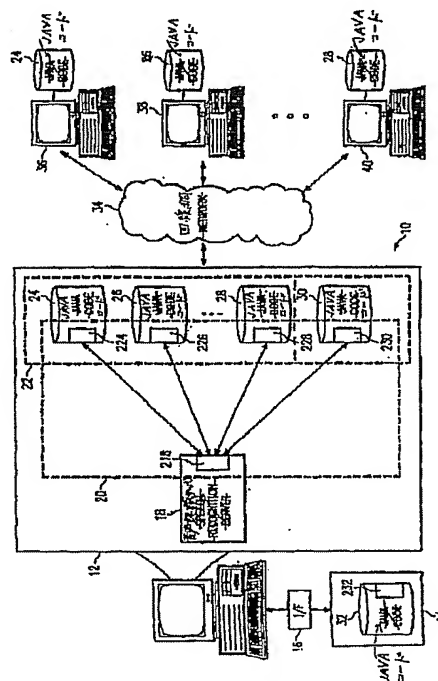
(74)代理人 弁理士 浅村 皓 (外3名)

(54)【発明の名称】 J A V A へ音声認識能力を加えるシステムと方法

(57)【要約】

【課題】 J a v a へ音声認識能力を加えるシステム
(10)。

【解決手段】 J a v a アプリケーション (24) は文
法 (56) をアプリケーションプログラムインターフェ
イス (20) へ動的に指定し、アプリケーションプログ
ラムインターフェイス (20) は、文法 (56) を音声
認識サーバ (18) へ通信する。音声認識サーバ (1
8) は、文法 (56) と音声入力 (66) を受信する。
音声入力 (66) を受信すると、音声認識サーバ (1
8) が音声認識を遂行して、文法 (56) に基づいた結
果 (58) を生成する。アプリケーションプログラムイ
ンターフェイス (20) が結果 (58) を J a v a アプリ
ケーション (24) へ通信し、J a v a アプリケーシ
ョン (24) は受信した結果 (58) に基づいて動作を
遂行する。



【特許請求の範囲】

【請求項1】 J a v aへの音声インターフェイスを容易にするシステムであって、
文法と音声入力を受け取るように作動可能な音声認識サーバであって、更に音声入力に応答して音声認識を遂行して、前記文法に基づいた結果を生成するように作動可能な前記音声認識サーバと、
動的に文法を指定し、前記結果を受信し、前記結果に基づいた動作を遂行するよう作動可能な少なくとも一つのJ a v aアプリケーションと、
文法を受信し、音声認識サーバへ文法を通信し、また音声認識に응答して、音声認識サーバから前記結果を受け取り、前記結果をJ a v aアプリケーションへ通信する、前記システム。

【請求項2】 J a v aへのインターフェイスを容易にする方法であって、
音声認識サーバとJ a v aアプリケーションの間に通信を設立するステップと、
音声認識サーバへ文法を通信するステップと、
音声入力に응答して音声認識を遂行し、文法に基づく結果を生成するステップと、
その結果をJ a v aアプリケーションへ通信するステップを含んでなる、前記方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 この発明は、一般にデータ通信および処理の分野に関し、特に、J a v aへ音声認識能力を加えるシステムと方法に関する。

【0002】

【従来の技術】 コンピュータのユーザは、長い間、口頭のコマンドに응答できるパーソナルなソフトウェアアプリケーションを要望してきた。入力媒体として音声を使用して、ユーザがソフトウェアと対話できるようにすれば、例えばキーボード、マウス、タッチスクリーン入力装置よりも、一層自然なインターフェイスが提供される。声の入力は、手を使わない(h a n d - f r e e)操作を容易にする長所を提供する。ソフトウェアアプリケーションとの対話中に、ユーザが他の事に対応可能になる上に、手を使わない操作は、身体障害者へ利用する機会を与える。声の入力は、スペルと構文の誤りの問題を避け、また他の入力方法に伴う大きなリストを通してスクロールせねばならないのを避けるという追加的な長所を有する。

【0003】 クライアントアプリケーションのための音声認識能力を供給する一つの方法は、HTML(ハイパーテキストマークアップランゲージ)に基づくスマートページの概念である。スマートページは、そのページに特有な文法へのリンクを含むワールドワイドウェブ(W e b)のページである。スマートページの著者は、そのページが응答すべき文法を定義し、スマートページ内に

一つのリンクを埋め込み、期待される言葉の入力のタイプに関する目に見えるキューを与える。音声エンジンがスマートページに出会うと、文法を組み込み、それが音声入力に응答できるようにし、結果をスマートページに返す。スマートページはこの結果を解釈して、これに従って응答する。

【0004】

【発明が解決しようとする課題】 このアプローチの不利な点は、HTMLに基づくW e bのページがステートレス(s t a t e l e s s)であること、すなわち、カレントのページから新しいページへリンクを追う時、新ページは前ページについて何も知らないことである。U R L(ユニフォームリソースロケータ)に状態情報をエンコードすることにより、この制限を克服できるが、この方法は非常に非効率な解決をもたらす。このアプローチの更に不利な点は、クライアントアプリケーション一般について、音声認識能力を加えるのに何の解決ももたらさないことである。HTMLは、完全なプログラミング言語ではないので、その実用的な利用は、W e bページとブラウズコマンドに制限される。

【0005】

【課題を解決するための手段】 この発明の教示によれば、J a v aへ音声認識能力を加えるシステムが提供されて、以前に開発されたシステムに関連する不利な点や問題を除去し、または著しく減少させる。

【0006】 この発明の一実施例によれば、J a v aへ音声認識能力を加えるシステムは、アプリケーションプログラムインターフェイスを通じてJ a v aアプリケーションに結合された音声認識サーバを含む。J a v aアプリケーションはアプリケーションプログラムインターフェイスへ文法を動的に指定し、アプリケーションプログラムインターフェイスはこの文法を音声認識サーバへ通信する。音声認識サーバは、文法と音声入力を受信する。音声認識サーバは、音声入力について音声認識を遂行して、文法に基づく結果を生成する。アプリケーションプログラムインターフェイスは、この結果をJ a v aアプリケーションインターフェイスへ通信して、J a v aアプリケーションは、受信した結果に基づく動作を遂行する。

【0007】 この発明は、J a v aアプリケーション内に状態情報を容易にエンコードする能力を含む重要な技術的長所を提供する。ステートレス(s t a t e l e s s)であるHTMLと異なり、J a v aは完全なプログラミング言語で、必要な状態情報を伝えることができる。その上、J a v aは完全なプログラミング言語なので、この発明は、あらゆるJ a v aプログラムのアプリケーションの音声作動を容易にし、W e bブラウジングのアプリケーションに限定されない。一つの更なる長所は、J a v aがプラットフォームから独立した言語であることである。それ自体として、この発明は、背景に音

声サーバを走らせれば、多重プラットフォーム上で同一の音声認識プログラムを使用可能にする。これにより、クライアントのプログラマは、オーディオレコーディングや音声認識装置の仕様のようなプラットフォームに從属した問題を無視できる。更にその上の長所は、Javaアプリケーションを音声可能にするために使用するアプリケーションプログラムの柔軟性である。この発明によれば、未経験のプログラマが単純なテンプレートを使用して、アプリケーションを敏速に音声可能にすることもできるが、また一方、より経験の深いプログラマに、より複雑な機能を実行する柔軟性を提供する。

【0008】この発明の更に一層の長所は、アプリケーションプログラムインターフェイスの基礎となるクライアント／サーバモデルである。音声認識サーバは、大量の処理負荷を取り扱うので、より遅く解読されるJavaアプリケーション上に、より軽い負荷が配置される。その上、このクライアント／サーバモデルは、独立の、恐らくはサーバコンピュータよりも力弱い装置上で、クライアントアプリケーションが実行可能にする柔軟性を有する。Web上でJavaプログラムと通信する時は、Javaのクライアント側の性質は、一つのインタラククション内の対話の文脈をたどることにより、大いに単純化される。その上、Javaアプリケーションとの直接の通信が、応答待ちの時の回線の遅延を除く。

【0009】この発明の更にもう一つの長所は、文法データ構造の動的な修正の提供である。動的修正は、ワールドワイドウェブサイトをブラウズするときのように、音声エンジンが会える文脈が予測できない場合に、価値ある長所である。こうした場合、動的修正は、出会ったアプリケーションの文脈に適合するように音声エンジンの言語を増補することを、音声認識サーバができるようにする。この発明の文法データ構成は、従来の単一正規文法にまさる簡潔さの追加的長所を有する。

【0010】

【発明の実施の形態】この発明の好ましい実施例を図1ないし図4に示すが、同一の参照番号は、種々の図面の同一の対応する部品を示す。

【0011】図1は、Javaを音声認識機能で作動にするシステムの単純化した表現である。このシステムは、Javaアプリケーション（アプリケーション）24、音声認識サーバ（SRS）18、サーバ側アプリケーションプログラムインターフェイス（サーバ側API）218、アプリケーション側アプリケーションプログラムインターフェイス（アプリケーション側API）224を含む。サーバ側API218およびアプリケーション側224は、集合的にAPI20と呼ばれる。作動中、アプリケーション24は、API20へ一つまたはそれ以上の文法を指定する。文法は、構文と語彙を定義する規則のセットで、アプリケーション24がそれに応答するものである。API20は、SRS18に文法

を通信し、SRS18は文法を受信し記憶する。文法を処理した後に、SRS18は音声入力を受信し、またJavaアプリケーション24により指定された文法に基づき、音声認識を遂行し、結果を生成する。Javaアプリケーション24は、受信された結果に基づき適切な動作を遂行する。

【0012】動作アプリケーション24は、アプリケーション24に指示されるコマンドに回答して取られる恐らくあらゆる動作を遂行する。この動作は、例えばコマンド「スクロールアップ」に回答したスクロールであり、コマンド「ゴージャック」に回答して以前のWebページへ戻ること、またはコマンド「予報を示せ」に回答して天気予報を表示することである。この動作は、結果を受信する特定のアプリケーションの機能により行われる。

【0013】図2は、音声ホストコンピュータ12を含む例示的なシステムを図示する。音声ホストコンピュータ12は、SRSおよびアプリケーション24ないし30を含み得る。アプリケーション24ないし32は、一般的にアプリケーション24と呼ばれ、APIを通じてSRS18とインターフェイスする。Javaアプリケーション24ないし28は、回線網34上に存在するネットワークコンピュータ36ないし38から起動する。Javaアプリケーション30は、メモリ内の音声ホストコンピュータ12上にローカルの存在し得るし、または音声ホストコンピュータ12上の据え付け部品としても存在し得る。API20は、アプリケーションインタプリタ22を通じて、Javaアプリケーション24、30とインターフェイスし得る。システム10は、さらにインタフェイス装置14を含み、インタフェイス装置14は、インタフェイス16を通じて音声ホストコンピュータ12へ結合される。Javaアプリケーション32は、インタフェイス装置14上に存在し得て、インタフェイス16を通じてAPI218と通信する。

【0014】音声ホストコンピュータ12は、メインフレームコンピュータ、ミニコンピュータ、ワークステーション、パーソナルコンピュータのようなSRS18およびAPI20をサポートできる何らかの装置であり得る。一般にSRS18は、クライアントアプリケーションからの接続を聴取し、クライアントサーバプロトコルを理解し、アプリケーション24により要求される動作を遂行する何らかの装置またはアプリケーションであり得る。より特定の実施例において、SRS18は話し手から独立した連続音声の、任意語彙の認識システムの中へ統合され得る。話し手からの独立は、新しい話し手が導入されたときに、新しい話し手の音声特性についての訓練無しに、認識の能力が機能し続けることを意味を言う。連続音性のサポートは、話された語の間の休止の必要無しに、認識器の能力が機能することを言う。任意語彙は、あらかじめ定めた語彙サイズおよびまたは内容

5

に制約されない語彙である。SRS18は、Javaアプリケーションだけに特定なものではない。それは、クライアントとサーバの間の通信のプロトコルに従うあらゆるアプリケーションとインタフェイスできる。

【0015】SRS18は、Netscape™または、Microsoft Explorer™のようなウェブブラウザと共に使用されて、ワールドワイドウェブとの音声対話のための特定の機能をサポートする。ワールドワイドウェブとの対話の時にいくつかの長所を提供することに加えて、SRS18は、種々の他のマルチメディア目的のための音声認識サーバとして使用できる。これらの中にはキヨスク、音声パワー付き（voice-powered）対話型マルチメディアプレゼンテーションを通じての情報アクセス、対話型訓練および修理マニュアルの音声ナビゲーション、ウェブを通じてのテレビジョンの統合がある。

【0016】アプリケーションプログラムインタフェース20は、ルーチン、コマンド、データ構造、バーチャル機能およびその他の機能、バーチャルおよび他の機能コール、データの定義、およびSRS18の一つまたはそれ以上の作動を制御する他の変数のコレクションを含み得る。API20はサーバ側のAPI218とアプリ

```
import javax.speech.* //The Java speech API package
...
public class MyApplet extends Applet implements Runnable, Speakable
{
    ...
    SpeechRecognizer SR;
    SRGrammar grammar;
    ...
    SR = new SpeechRecognizer(this);
    if (SR.connect()) {
        SRGrammar grammar = new SRGrammar("blink_grammar");
        grammar.addRule("blink_grammar ---> start blinking | stop blinking");
        SR.loadGrammarToRecognizer(grammar);
        SR.startListening();
    }
    ...
    public void onSpeechRecognition(SRResult result)
    {
        // simulated callback that receives the SRResult object and is
        // responsible for its processing
    }
    ...
    SR.stopListening();
    SR.disconnect();
    ...
}
```

【0018】プロジェクト指向プログラミングにおいては、一つのクラスが、属性、サービス、データ構造、ルーチン、またはクラス内またはクラスのインスタンスのオブジェクトを特徴付ける他の情報を定義する。上記の例示的なコードにおいては、使用されたクラスは、SpeechRecognizer、SRResult、S

6

*ケーション側の複数のAPI224ないし232を含むが、一般的にAPI224と呼ばれる。サーバ側のAPI218は、SRS18の作動を制御するのに必要なコードの大部分を含む得る。アプリケーション側のAPI224は、SRS18とアプリケーション24の間の対話を容易にする高度の特定の行を含む。サーバ側のAPI218およびアプリケーション側のAPI224は、集合的にAPI20を形成する。API20は、特定のアプリケーションに独自の文法仕様と指示（directive）がSRS18の部分になることを可能にする。この機能を使用して、プログラマはアプリケーション側のAPIを追加することにより、いずれかのJavaアプリケーションを音声可能にでき、こうしてSRS18とアプリケーションの間の対話を容易にできる。この実施例において、アプリケーション24ないし32は、それらにアプリケーション側のAPI224ないし232をそれぞれエンコードすることにより、音声可能にできた。

【0017】例示の方法により、限定としてではなく、下記の例示的なコードをアプリケーション24を音声可能にするために使用できる。

【表1】

```
import javax.speech.* //The Java speech API package
```

```
...
public class MyApplet extends Applet implements Runnable, Speakable
{
```

```
    ...
    SpeechRecognizer SR;
    SRGrammar grammar;
    ...
    SR = new SpeechRecognizer(this);
    if (SR.connect()) {
        SRGrammar grammar = new SRGrammar("blink_grammar");
        grammar.addRule("blink_grammar ---> start blinking | stop blinking");
        SR.loadGrammarToRecognizer(grammar);
        SR.startListening();
    }
    ...
    public void onSpeechRecognition(SRResult result)
    {
        // simulated callback that receives the SRResult object and is
        // responsible for its processing
    }
    ...
    SR.stopListening();
    SR.disconnect();
    ...
}
```

RGrammarである。SpeechRecognizerのクラスは、SRS18とアプリケーション24の間に接続を設定したり、SRS18と通信したり、SRS18との接続を切断したりするような基本的な音声認識の作動を制御する。SRGrammarのクラスは、文法56を指定するのに使用される。SRGram

marを通じて、アプリケーション24は、アプリケーション24の実行中に、文法を追加した置き換え、文法を停止した再起動し、動的に文法を再格納した変更する。SRResultのクラスは、文法56、文法56から認識される結果58、一つの信頼得点(confidence score)およびn個の最良の結果のリストとその信頼得点、のような情報を含有できる。非同期認識のために必要とされる疑似コールバックを扱うために、アプリケーション24は、機能onSpeechRecognitionをも含み得るが、これは認識結果58が受信された時に自動的に呼出される。

【0019】上記の例示的なコードは、極度に簡単なテンプレートにより音声でアプリケーション24を実行可能にする方法を示す。例示コードの単純さは、音声認識APIに未経験な人々が、最小の努力でアプリケーションを音声で実行可能にするという長所をもたらす。API20は、より経験が深いプログラマが望むような、進歩した音声認識能力をサポートする付加的な長所を有する。

【0020】アプリケーション24は、Javaプログラミング言語にふさわしいどんなアプリケーションにも適する。例として、限定としてではなく、アプリケーション24は、コンピュータバンキングプログラム、コンピュータゲームプログラム、またはワールドワイドウェブのページであり得る。Javaは、プラットフォームから独立したプログラミング言語である。それ自体で、この発明は、音声ホストコンピュータ12のプラットフォームに関係なく、音声認識機能付きのクライアントアプリケーションを実行可能にする長所を提供する。

【0021】API20は、多数の方法でアプリケーション24ないし32とインタフェイスし得る。特定の実施例において、アプリケーション30は、音声ホストコンピュータ12に対してローカルであり得る。たとえばアプリケーション30は、音声ホストコンピュータ12のハードドライブ上に存在し得るし、フロッピードライブまたはCD-ROMのような外部ドライブ上に存在し得るし、または音声ホストコンピュータ内へハードワイヤされる。一般にアプリケーション30は、音声ホストコンピュータ12の自局側に、アプリケーション30を記憶するのにふさわしいいずれかの媒体に存在して、システム10内におけるその作動を容易にする。

【0022】もう一つの実施例においては、アプリケーション32は携帯電話、パーソナルディジタルアシスタント、またはセットトップボックスのようなインタフェイス装置14上に存在する。アプリケーション32は、インタフェイス16を通じてAPI20と通信し得る。インタフェイス16は、インタフェイス装置14と音声ホストコンピュータ12の間の通信を容易にするのにふさわしいいずれかの有線または無線装置を含み得る。この実施例において、処理負荷の大部分は音声ホストコン

ピュータ12上に置かれるので、クライアントアプリケーション32はパワーとメモリの大きな容量を持つ必要がない。これにより、さもなければ必要なメモリとパワーの要件を欠くような、小さなパワーの弱いインタフェイス装置上で走るアプリケーションに、音声認識機能を提供するという長所をもたらす。

【0023】さらにもう一つの実施例において、アプリケーション24ないし28は、回線網34上のどこかに存在し得る。回線網34は、いずれかの適当な有線または無線のリンクを含み得て、音声ホストコンピュータ12と回線網コンピュータ36ないし40の間の通信を容易にする。たとえば回線網34は、公衆交換電話網、統合サービスデジタル回線網(ISDN)、ローカルエリアネットワーク(LAN)、首都圏ネットワーク(MAN)、広域ネットワーク(WAN)、ワールドワイドウェブのようなグローバルネットワーク、または他の専用または交換回線網または他の通信機構を含み得る。音声ホストコンピュータ12は、アプリケーションインタプリタ22を通じて、アプリケーション24ないし32とインタフェイスできる。アプリケーションインタプリタ22は、回線網34にアクセスするのにふさわしいいずれかの装置であり得て、ブラウザまたはアプレット(applet)ビューのようなアプリケーション24を発見し検索するのにふさわしい装置であり得る。

【0024】図3は、この発明の教示による音声認識サーバ18と、Javaアプリケーション24ないし32の間の、例示的なインタフェイスである。音声ホストコンピュータ12は、SRS18、API20および一般にアプリケーション24と呼ばれている一つまたはそれ以上のJavaアプリケーション24ないし32をサポートする。SRS18は音声エンジン50を含み、これは文法データ構造52およびアプリケーションデータ構造54を有する。一般に音声エンジン50は音声入力に際しての音声認識を遂行するのにふさわしいいずれかのアルゴリズムを含み得る。特定の実施例において音声エンジン50は、当分野で知られているように少なくとも3つの層を含む。すなわち音響モデルを表現する隠れマルコフモデル層、オプションの単語発音層、予測される語のシーケンスを指定するセンテンスシンタックス層を含み得る。

【0025】一般に文法データ52は、いずれかのデータ記憶装置中、および文法56の検索を受信し記憶し容易にするために作動できるアプリケーション中に、記憶され得る。特定の実施例において、文法データ構造52は、当分野で知られているような、閉路なし有向グラフにおける正規文法(RGDAG)であり得る。RGDAGにおいて、第1文法における終末記号が第2文法における開始記号に一致する時は、一つの文法はもう一つに接続する。文法データ構造52としてRGDAGを実施すれば、音声エンジン50がRGDAG内の開始信号の

いずれかのサブセットにより、その探索プロセスを開始できる。これは、出会ったアプリケーションの文脈による、音声エンジンの言語の動的な修正を容易にする。

【0026】RGDAGの動的修正は、ワールドワイドウェブサイトのブラウジングのように、音声エンジンが出会う文脈が予測不可能な状況において、音声認識サーバ18の一つの長所である。こうした場合において、文法の動的修正は、音声認識サーバ18が音声エンジン50の言語を増補して、出会ったアプリケーションの文脈に適合させることを可能にする。RGDAG構造は、従来の単一正規文法よりも一層効率的な文法記憶装置を容易にすることにより、簡潔さの付加的な長所を提供する。

【0027】アプリケーションデータ構造54は、アプリケーション24の跡を追う。アプリケーションデータ構造54は、アプリケーション24を「アクティブ」または「アクティブでない」として、「聞いている」または「聞いていない」として、または「実行中」または「実行中でない」と指定し得る。付加的にアプリケーションデータ構造54は、各文法を「動作可能」または「不動作」としてマークすることにより、各アプリケーションに結びついた文法の状態を追跡できる。この方法で、音声認識サーバ18は、多数のクライアントアプリケーションの同時的な作動を追跡し、サポートできる。アプリケーションデータ構造は、更にアプリケーション24が音声焦点(speech focus)を受け取れるようにすることができる。音声焦点は自身の文法からの結果のために各々聞いている、多数のアプリケーションの間の競合を解決するのに、必要であり得る。アプリケーション24に音声焦点を与えることにより、アプリケーション24は、音声エンジン50が他の聞き取りアプリケーションへ結果58を送信することを、防止できる。この方法で目的のアプリケーションであるアプリケーション24が、結果58を受信する。

【0028】SRS18は、またAPI20のサーバ側部分、サーバ側API218を含み得る。サーバ側API218は、パーサ60を含み得る。パーサ60は、音声エンジン50から結果58を受信し、この結果を解析して、解析結果61を、アプリケーション24へ通信する。作動においてパーサ60は、クライアント24が文法56内の一定の非終端子を指定するのを許容する。パーサ60は、それから指定された非終端子に対応する結果58の部分のみを、アプリケーション24へ返す。例えば文法56は形式: <冠詞> <形容詞> <名詞> <副詞> <動詞> <前置詞句> であり得るし、パーサ60は名詞と動詞を解析結果61として抽出し、解析結果61のみをアプリケーション24へ渡す。これによりアプリケーション24は、不必要な情報を無視できる。パーサ60の一つの長所は、アプリケーション24が結果58を再解析する必要を除くことであり、この

再解析は、複雑な文法についてわずらわしいプロセスであり得る。

【0029】サーバ側API218はさらに、パラメタイザ(parametizer)62をさらに含み得る。パラメタイザ62は、アプリケーション24からパラメータ64を受信して、音声エンジン50へパラメータ64を通信し得る。音声エンジン50は、パラメータ64を受信してその作動を相応に修正し得る。パラメータ64は音声エンジン50の関数内のいずれかの変数であり得て、たとえばエンドポイントティングパラメータ、すなわち音声の発言が終了したことを決定するのに充分な沈黙の持続を指定するものである。

【0030】図4も参照すると、音声可能アプリケーション24のための例示的な方法が示されている。この方法はステップ100で始まり、ここでアプリケーション24とSRS18の間の接続が設定される。この接続が設定されると、ステップ102でSRS18が、アプリケーションデータ構造54の中のアクティブアプリケーションのリストへアプリケーション24を追加し、周期的にアプリケーション24からのコマンドをチェックする。アプリケーション24は、それからステップ104でAPI20への文法を動的に指定する。文法56は、アプリケーション24がAPI20へのそのアクセスおよび指定を許すあらゆる位置または状態に存在し得る。たとえば文法56は、アプリケーション24のJavaコード内のテキストストリング内、アプリケーション24により指定されたファイル内、またはアプリケーション24により指定されたユニフォームリソースロケータ(URL)内に存在し得る。

【0031】更に加えて、ステップ104においてアプリケーション24は、パラメータ64を指定し得る。上記のようにパラメータ64は、音声エンジン50のあらゆる変数に対応し得る。API20は、それから文法56とパラメータ54をステップ105でSRS18へ通信する。またステップ104でアプリケーション24は、すでにSRS18へ通信された文法が「実行可能にされた」または「実行不可能にされた」かを指定し得る。

【0032】ステップ106で、SRS18が文法56を受け取ると、それは文法56を文法データ構造52の中のアプリケーション24に関連する文法のリストに加える。またステップ106においてSRS18は、アプリケーション24により指定されたように文法データ構造52の中の文法を、SRS18が実行可能にし、または実行不可能にする。さらにSRS18は、パラメータ64の値によって音声エンジン50のパラメータを調節し得る。

【0033】ステップ107でもしアプリケーション24がすでに「リスニング」であれば、ステップ108で「リスニング」信号をAPI20を通じてSRS18へ

通信する。ステップ109で「リスニング」信号を受け取ると、SRS18はアプリケーション24をアプリケーションデータ構造54内のリスニングアプリケーションのリストへ加える。このアプリケーションが「リスニング」としてマークされると、SRS18はアプリケーション24にステップ110で音声焦点(speech focus)を与える。上記のように音声焦点は、音声エンジン50が結果58を他のリスニングアプリケーションへ送信するのを防止する。

【0034】ステップ111で音声エンジン50は音声入力66を受信し、ステップ112で音声認識を遂行し、音声入力66に基づいた結果58を生成し得る。ステップ113でパーサ60は、結果58を受信し、結果58を解析し得る。ステップ114でそれからパーサ60は、解析された結果61をアプリケーション24へ通信し得る。ステップ116で結果58または解析された結果61を受け取ると、アプリケーション24は受け取った結果に基づく動作を遂行し得る。

【0035】ステップ118でアプリケーション24は、追加の文法および/またはパラメータを指定し、およびまたは文法データ構造52内の文法を実行可能または実行不可能にし得る。代案として、ステップ120でアプリケーション24は、API20へ「ストップリスニング」信号を通信することにより、一時的に通信を中断できる。ステップ121で「ストップリスニング」の信号を受け取ると、SRS18はアプリケーションデータ構造54内のリスニングアプリケーションのリストからアプリケーション24を除く。ステップ122でアプリケーション24は、追加の文法を送り、および/またはAPI20へ「リスニング」信号を通信することにより、SRS18と通信を再開する。代案として、ステップ124でAPI20へ「切断」信号を通信することにより、アプリケーション24はSRS18との通信を終了できる。ステップ126で「切断」信号を受け取ると、SRS18はアプリケーションデータ構造からアプリケーション24を除去する。

【0036】この発明とその長所を詳細に説明してきたが、前記の特許請求の範囲により定義されたこの発明の精神と範囲から離れることなく、種々の変更、置き換え、改変がそこになし得ることを理解すべきである。

【0037】以上の説明に関してさらに以下の項を開示する。

【0038】(1). Javaへの音声インターフェイスを容易にするシステムであって、文法と音声入力を受け取るように作動可能な音声認識サーバであって、更に音声入力に応答して音声認識を遂行して、前記文法に基づいた結果を生成するように作動可能な前記音声認識サーバと、動的に文法を指定し、前記結果を受信し、前記結果に基づいた動作を遂行するよう作動可能な少なくとも一つのJavaアプリケーションと、文法を受信し、

音声認識サーバへ文法を通信し、また音声認識に応答して、音声認識サーバから前記結果を受け取り、前記結果をJavaアプリケーションへ通信する、前記システム。

【0039】(2) 前記音声認識サーバは、音声認識を遂行するよう作動可能な音声エンジンと前記文法を受信し記憶するよう作動可能な文法データ構造を含んでなる第(1)項記載のシステム。

【0040】(3) 前記音声認識サーバは、音声認識を遂行し結果を生成するよう作動可能な音声エンジンと前記文法を受信し記憶するよう作動可能な文法データ構造と、Javaアプリケーションの状態を追跡し、またJavaアプリケーションに関連した文法の状態を追跡するように作動可能なアプリケーションデータ構造を含んでなる第(1)項記載のシステム。

【0041】(4) 前記アプリケーションデータ構造は、結果の通信を目的とするJavaアプリケーションへ制限するように作動可能な音声焦点(speech focus)を含んでなる第(3)項記載のシステム。

【0042】(5) 前記音声認識サーバは、音声認識を遂行し、結果を生成するよう作動可能な音声エンジンとJavaアプリケーションの状態を追跡し、Javaアプリケーションに関連する文法の状態を追跡するように作動可能なアプリケーションデータ構造と前記文法を受信した前記文法の受信に応答して、閉路なし有向グラフ中の正規文法のセットの内容を動的に修正するよう作動可能な、閉路なし有向グラフ中の正規文法のセットを、含んでなる請求項1のシステム。

【0043】(6) 前記アプリケーションデータ構造は結果の通信を目的のJavaアプリケーションに限定するよう作動可能な音声焦点(speech focus)を含んでなる第(5)項記載のシステム。

【0044】(7) アプリケーションプログラムインタフェイスは、サーバ側アプリケーションプログラムインタフェイスとアプリケーション側アプリケーションプログラムインタフェイスを含んでなる第(1)項記載のシステム。

【0045】(8) アプリケーションプログラムインタフェイスは、サーバ側アプリケーションプログラムインタフェイスとアプリケーション側アプリケーションプログラムインタフェイスを含んでなり、サーバ側アプリケーションプログラムインタフェイスは、音声認識サーバからの結果を受信し、前記結果を解析し、解析された結果を生成するよう作動可能であり、前記サーバ側アプリケーションプログラムインタフェイスは、さらに解析された結果をJavaアプリケーションへ通信するよう作動可能な第(1)項記載のシステム。

【0046】(9) アプリケーションプログラムインタフェイスは、サーバ側アプリケーションプログラムインタフェイスとアプリケーション側アプリケーションプ

ログラムインタフェイスを含んでなりサーバ側アプリケーションプログラムインタフェイスは、Javaアプリケーションから一つのパラメータ値を受信し、また受信した前記パラメータ値により、音声認識サーバの一つのパラメータを修正するように作動可能なパラメタイザ (parameterizer) を含んでなる第(1)項記載のシステム。

【0047】(10) 音声認識サーバは、任意のサイズを有する語彙を含んでなり、音声認識サーバは、話し手から独立しており、連続音声をサポートする、第(1)項記載のシステム。

【0048】(11) Javaへのインタフェイスを容易にする方法であって、音声認識サーバとJavaアプリケーションの間に通信を設立するステップと、音声認識サーバへ文法を通信するステップと、音声入力にตอบสนองして音声認識を遂行し、文法に基づく結果を生成するステップと、その結果をJavaアプリケーションへ通信するステップを含んでなる、前記方法。

【0049】(12) 音声認識サーバとJavaアプリケーションの間の通信を設定するステップは、Javaアプリケーションへアプリケーションデータ構造を追加するステップを含んでなる第(1)項記載の方法。

【0050】(13) 一つの文法を前記音声認識サーバへ通信するステップは、前記文法をアプリケーションプログラムインタフェイスへ指定するステップと、前記文法を一つの文法データ構造へ追加するステップと、前記文法の前記追加にตอบสนองして前記文法データ構造を動的に修正するステップを含んでなる、第(1)項記載の方法。

【0051】(14) 音声認識サーバへ一つの追加的な文法を通信するステップをさらに含んでなる第(1)項記載の方法。

【0052】(15) 前記Javaアプリケーションに通信される以前に結果を解析するステップをさらに含んでなる第(1)項記載の方法。

【0053】(16) パラメータ値を受信するステップと前記パラメータ値に基づいて音声認識サーバのパラメータを修正するステップをさらに含んでなる第(1)項の方法。

【0054】(17) 目的とするJavaアプリケーションに一つの音声焦点 (speech focus) を与えるステップをさらに含んでなる第(1)項記載の方法。

【0055】(18) Javaアプリケーションへの音声インタフェイスを容易にする方法であって、音声認識サーバとJavaアプリケーションの間の通信を設定するステップと、アプリケーションプログラムインタフェイスへ文法を指定するステップと、音声認識サーバへ文法を通信するステップと、文法の受信にตอบสนองして、閉路なし有向グラフ中の一組の正規文法を動的に修

正するステップと、音声入力にตอบสนองして音声認識を遂行し、文法に基づく結果を生成するステップと、その結果をJavaアプリケーションへ通信するステップを含んでなる、前記方法。

【0056】(19) 音声認識サーバへパラメータ値を指定するステップ、および受信したパラメータ値に基づき音声認識サーバのパラメータ値を修正するステップと、音声認識サーバにより生成された結果を解析するステップ、および解析された結果を生成するステップ、および解析された結果をJavaアプリケーションへ通信するステップと、音声焦点 (speech focus) をJavaアプリケーションに与えるステップを含んでなる、第(1)項記載の方法。

【0057】(20) 多数のJavaアプリケーションの状態を追跡するステップと、Javaアプリケーションに関連する文法を追跡するステップを、更に含んでなる第(1)項記載の方法。

【0058】(21) Javaへ音声認識能力を加えるシステム10において、アプリケーションプログラムインタフェイス20を通じてJavaアプリケーション24へ結合された音声認識サーバ18が提供される。Javaアプリケーション4は文法56をアプリケーションプログラムインタフェイス20へ動的に指定し、アプリケーションプログラムインタフェイス20は、文法56を音声認識サーバ18へ通信する。音声認識サーバ18は、文法56と音声入力66を受信する。音声入力66を受信すると、音声認識サーバ18が音声認識を遂行して、文法56に基づいた結果58を生成する。アプリケーションプログラムインタフェイス20が結果58をJavaアプリケーション24へ通信し、Javaアプリケーション24は受信した結果58に基づいて動作を遂行する。

【図面の簡単な説明】

この発明のより良き理解のために、添付の図面を参照し得る。

【図1】この発明の教示により、Javaアプリケーションへの音声インタフェイスを容易にするための単純化された例示的なシステムを図示する。

【図2】この発明の教示により、Javaアプリケーションへの音声インタフェイスを容易にするための例示的なシステムを図示する。

【図3】この発明の教示による音声認識サーバとJavaアプリケーションの間の例示的なインタフェイスを図示する。

【図4】この発明の教示により、Javaアプリケーションへの音声インタフェイスを容易にするための例示的な方法を図示する。

【符号の説明】

10 システム

12 音声ホストコンピュータ

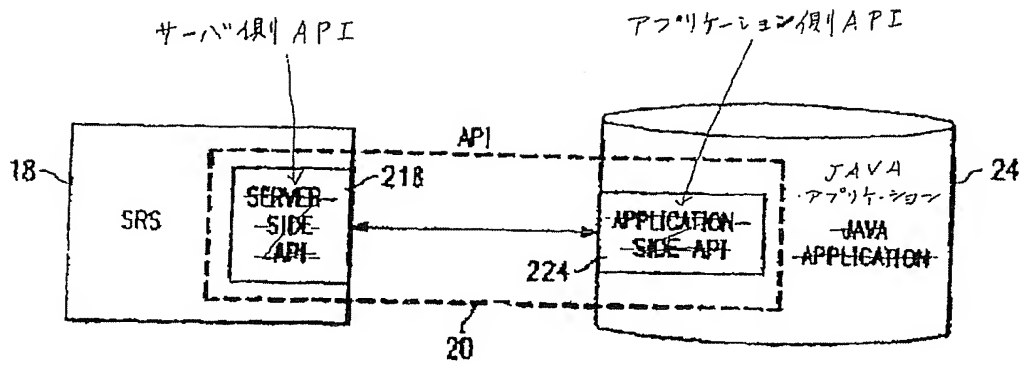
15

16

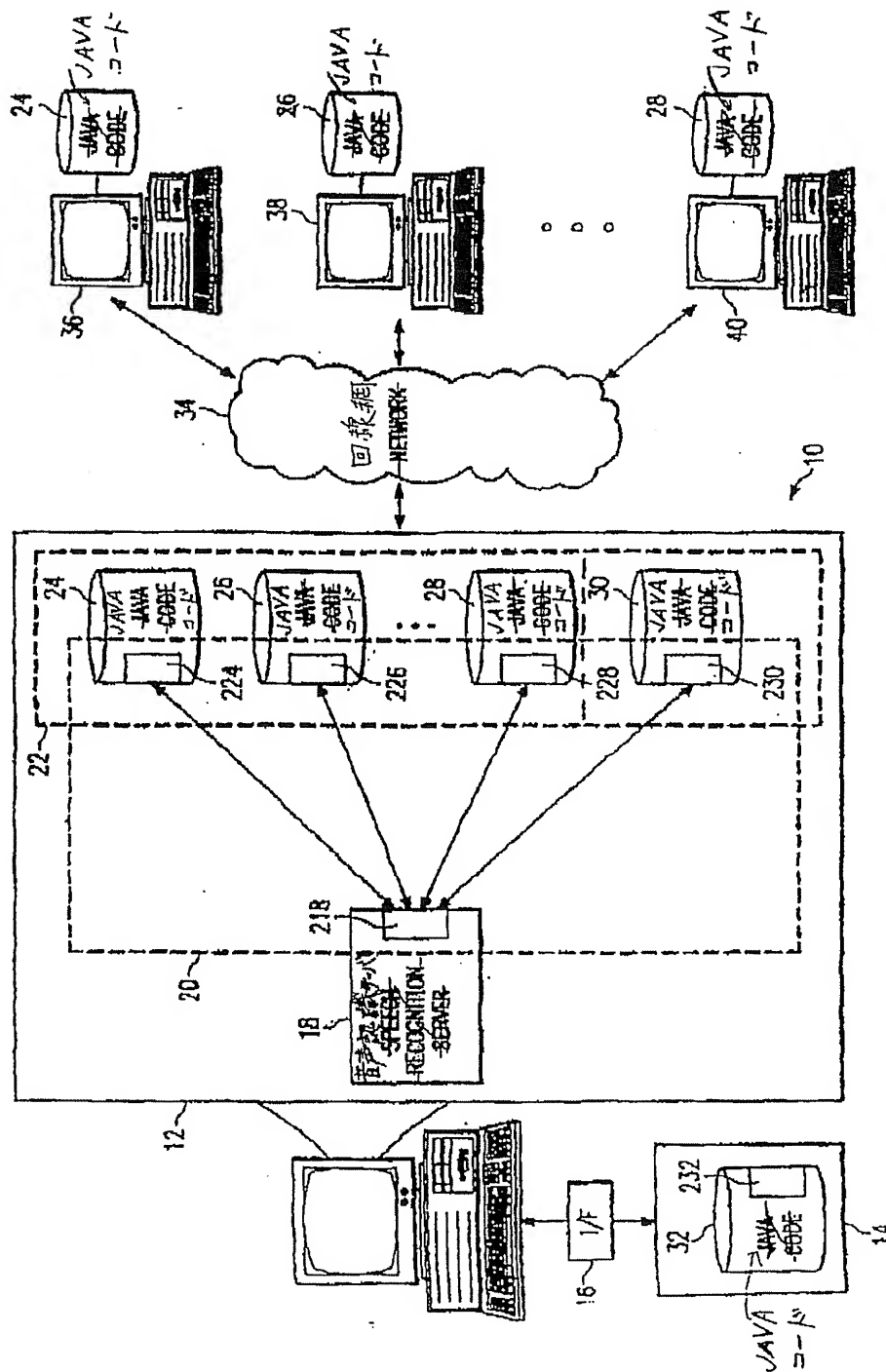
18 音声認識サーバ
 20 アプリケーションプログラムインターフェイス
 24 Javaアプリケーション
 56 文法

58 結果
 60 パーサ
 66 音声入力

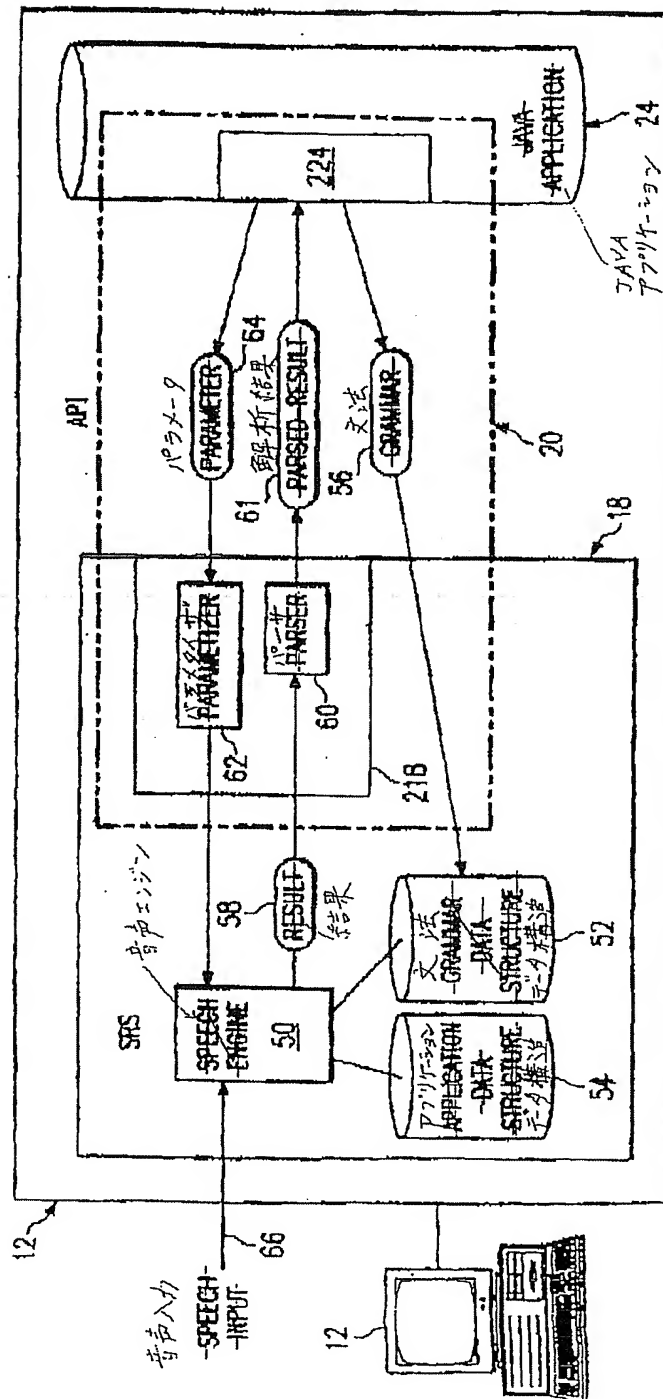
【図1】



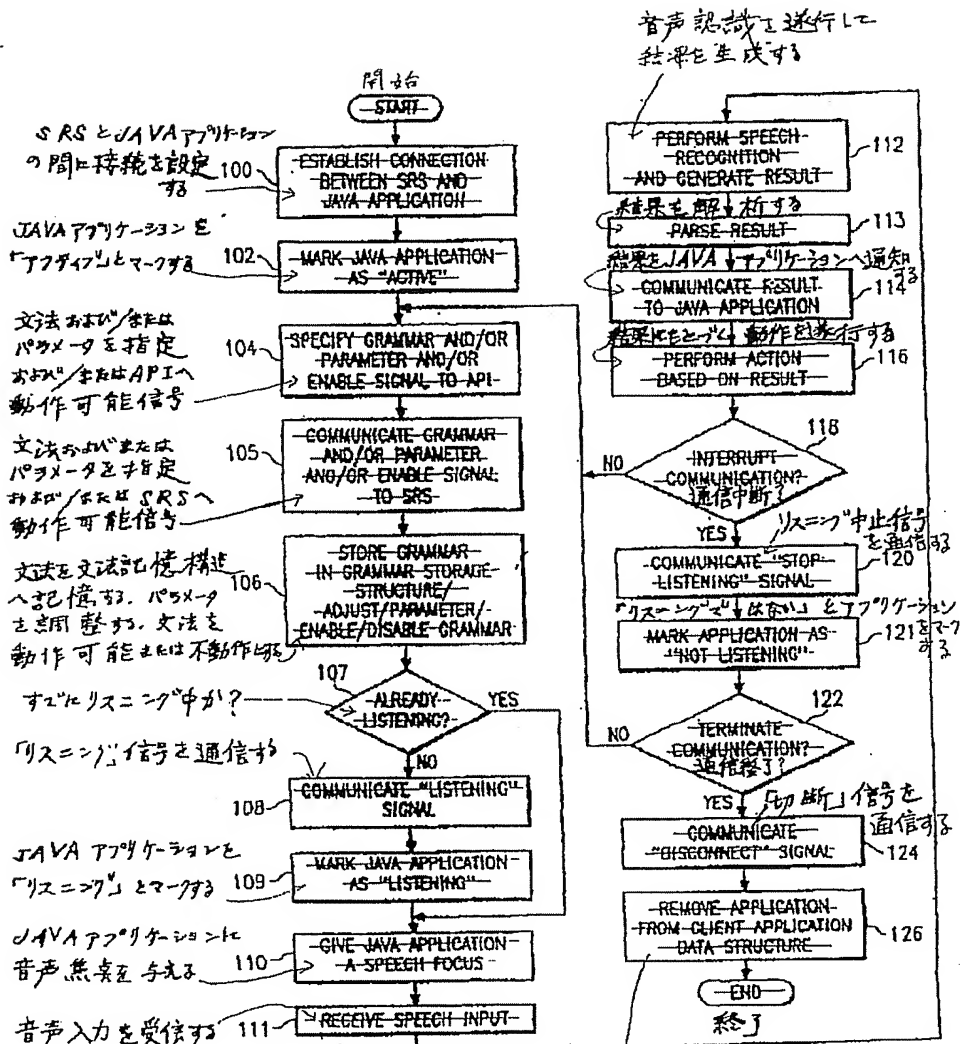
【図2】



- 11 -



【図4】



クライアントアプリケーションデータ
構造からアプリケーション
を削除する